PP4AV: A benchmarking Dataset for Privacy-preserving Autonomous Driving Supplementary Material

Linh Trinh[✉1] , Phuong Pham[1], Hoang Trinh[1], Nguyen Bach[1],
Dung Nguyen[2], Giang Nguyen[1], and Huy Nguyen[1]

[1]VinFast, Hanoi, Vietnam
v.{linhtk13, phuongpnv, hoangtn, nguyenbh1, giangnd2, vantix01}@vinfast.vn
[2]FPT Software, AI center, Vietnam dungnt20@fsoft.com.vn

1. PP4AV Dataset

1.1. Annotation process

We propose an annotation process which consists of two steps, as illustrated in the Figure 1. In the first step, two annotators will work independently on the same sets of images. Once completed, their annotation output will be go through a merging procedure, based on the IoU scores between the 2 bounding boxes of the 2 annotations. Annotations that are in pairs and have IoU scores over a threshold will be merged into one and saved as a merged annotation. A pair of annotations is considered conflicting if the IoU score is less than the cutoff. In the second step, two distinct reviewers will review this pair of conflicting annotations for correction, before the second merging procedure is carried out. This stage's merging procedure is the same as the first step. To acquire the final annotation, the merged revision from the second step will be concatenated with the merged annotation result from the first step.

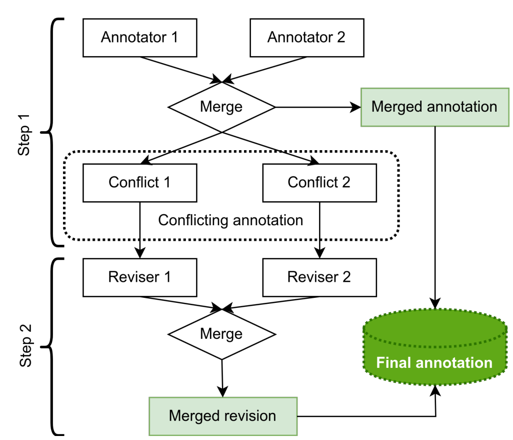


Figure 1: Illustration of annotation process.

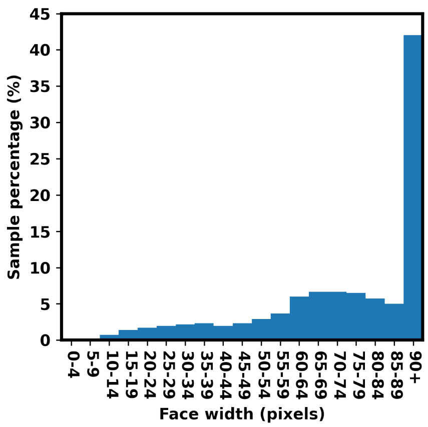
1.2. Histogram of object size

The Figure 2 show the comparison of histogram of face width between PP4AV dataset with AFW, UFDD, FDDB and WIDER FACE dataset. AFW indicates that the annotated faces are extremely large, with face widths exceeding 90 pixels. In the FDDB dataset, nearly 40% of facial objects exceed 90 pixels in width, which is larger than the PP4AV dataset. Similar to the PP4AV dataset, UFDD and WIDER FACE provide a wide variety of face widths, the majority of which are small. Nevertheless, these two datasets contain facial objects with a large size (greater than 90 pixels) because they were collected from a variety of natural scenes. The majority of faces in the street scene have widths between 5 and 40 pixels, as indicated by the histogram of PP4AV. It demonstrates the necessary requirements for the development of a benchmark dataset for face detection in driving scenarios, with a strong emphasis on small object sizes in street scenes.

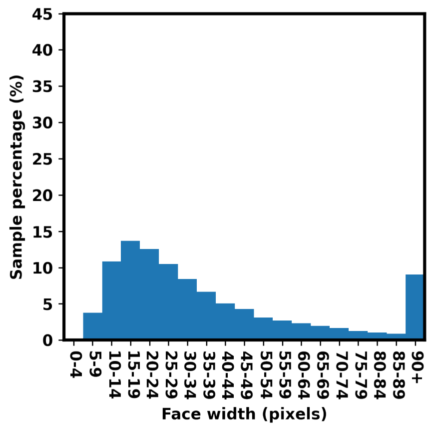
The Figure 3 show the comparison of histogram of plate height between PP4AV dataset with Lucian, SSIG-SegPlate, UFPR-ALPR and CCPD dataset. More than 50% of the plate heights in the CCPD dataset are greater than 90 pixels, whereas the majority of plate heights in the UFPR-ALPR and Lucian datasets fall between 20 and 60 pixels. The height of license plates in the SSIG-SegPlate dataset ranges from 25 to 60 pixels. In the PP4AV dataset, the plate height of both normal camera and fisheye images is typically less than 20 pixels. Due to the smaller size of license plates in traffic scenes compared to those captured in parking spaces, the PP4AV dataset presents a significant challenge for license plate detection.

2. Experimental results and analysis

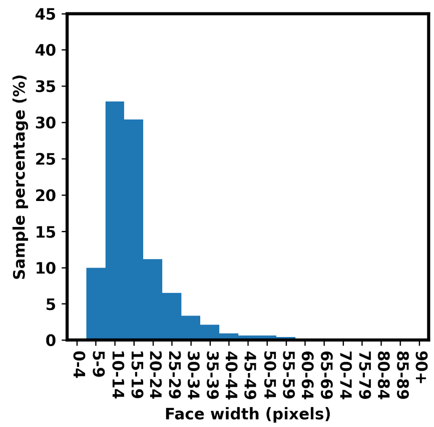
Comparison of performance between our model with other methods. Table 1 compares the detection perfor-



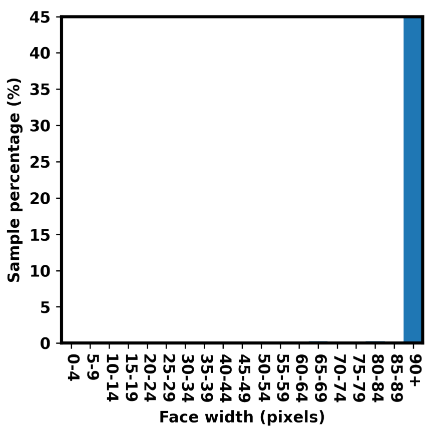
(a) FDDB



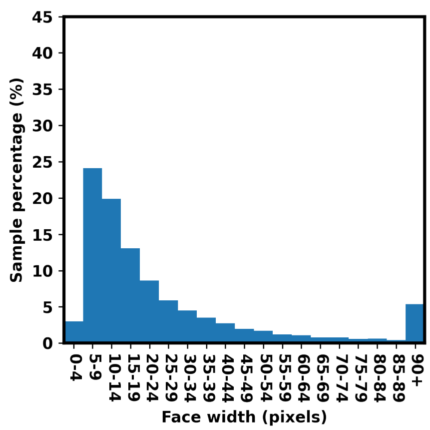
(c) UFDD



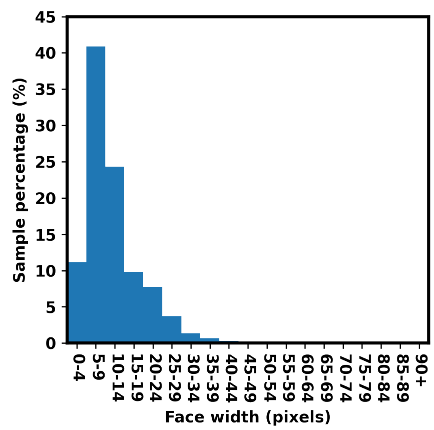
(e) PP4AV normal images



(b) AFW



(d) WIDER FACE



(f) PP4AV fisheye images

Figure 2: Comparison of histogram of face width between (e) normal camera images and (f) fisheye images of PP4AV with (a) FDDB, (b) AFW, (c) UFDD, and (d) WIDER FACE.

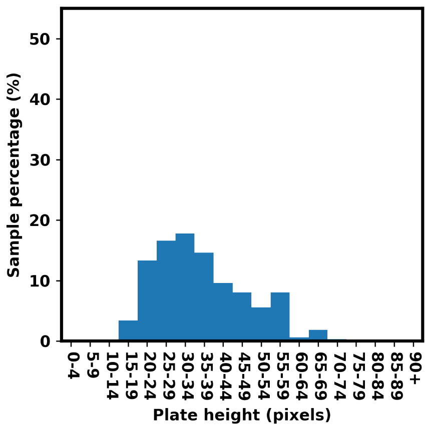
mance of faces and license plates in normal and fisheye photos from the PP4AV dataset. In this table, all detection algorithms are evaluated without filtering based on object size. Our model demonstrates that it rank at the top for face and license plate detection on normal camera images together with YOLO5Face for face detection and UAI Anonymizer for license plate detection. Due to the fact that it is not designed for fisheye images, our model performs less well than YOLO5Face in detecting faces in fisheye images. But our model still work well as UAI Anonymizer on plate detection on fisheye images.

The Figure 4 show the ratio of detection object counts and ground-truth object counts versus the face width and plate height for different methods.

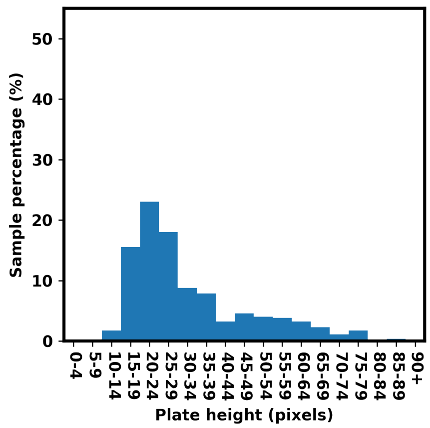
Qualitative examples. The Figure 5, 6, 7 below shows the example of detection results of different methods on face and license plate on PP4AV dataset.

	Methods	Normal images		Fisheye images	
		AP_50	AR_50	AP_50	AR_50
Face	UAI Anonymizer	42.64%	83.65%	44.15%	53.33%
	AWS API	69.64%	79.93%	52.44%	58.52%
	Google API	7.97%	8.99%	7.64%	8.89%
	RetinaFace	62.84%	88.79%	43.82%	62.96%
	YOLO5Face	85.76%	**92.54%**	**74.37%**	**88.15%**
	Our.	**86.11%**	91.47%	44.15%	53.38%
Plate	ALPR	42.8%	45.14%	18.12%	30.63%
	NVIDIA LPnet	52.85%	53.39%	28.13%	28.12%
	UAI Anonymizer	91.85%	92.78%	**58.78%**	61.25%
	Our.	**92.34%**	**93.54%**	56.36%	**63.9%**

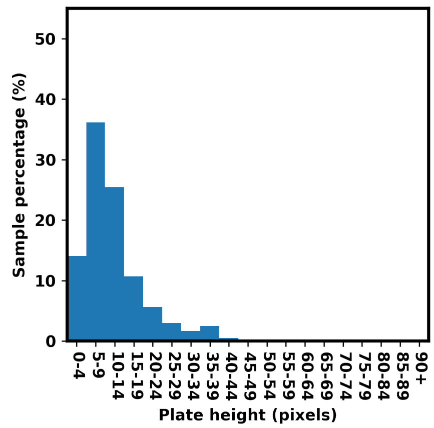
Table 1: Average Precision (AP) and Average Recall (AR) scores corresponding to different methods on our dataset ('-': model have no detection).



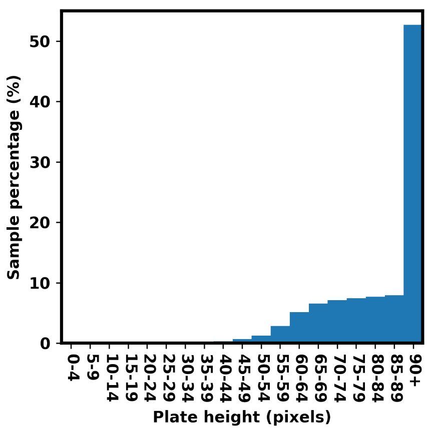
(a) UFPR-ALPR



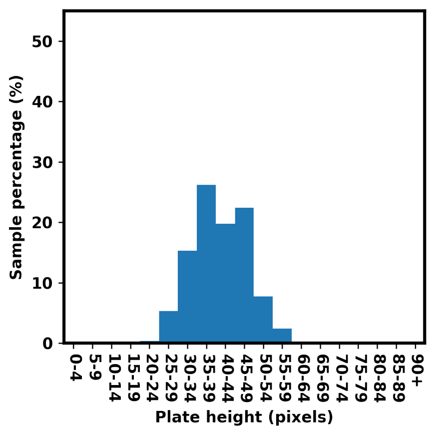
(c) Lucian



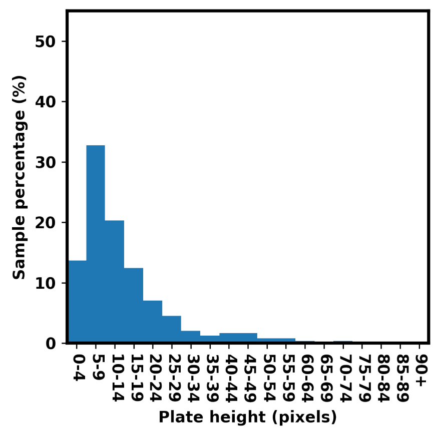
(e) PP4AV normal images



(b) CCPD

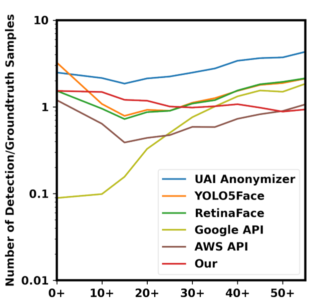


(d) SSIG-SegPlate

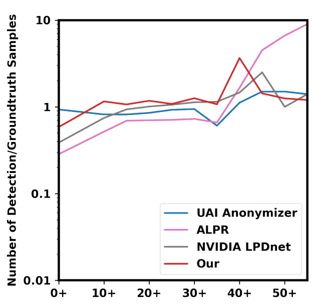


(f) PP4AV fisheye images

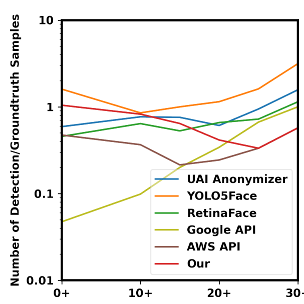
Figure 3: Comparison of histogram of license plate height between (e) normal camera images and (f) fisheye images of PP4AV with (a) UDFD-ALPR, (b) CCPD,(c) Lucian and (d) SSIG-SegPlate.



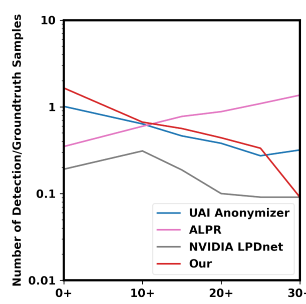
(a) PP4AV normal images



(b) PP4AV normal images



(c) PP4AV fisheye images



(d) PP4AV fisheye images

Figure 4: Ratio of detection and ground-truth object counts versus object size for different methods on PP4AV dataset.

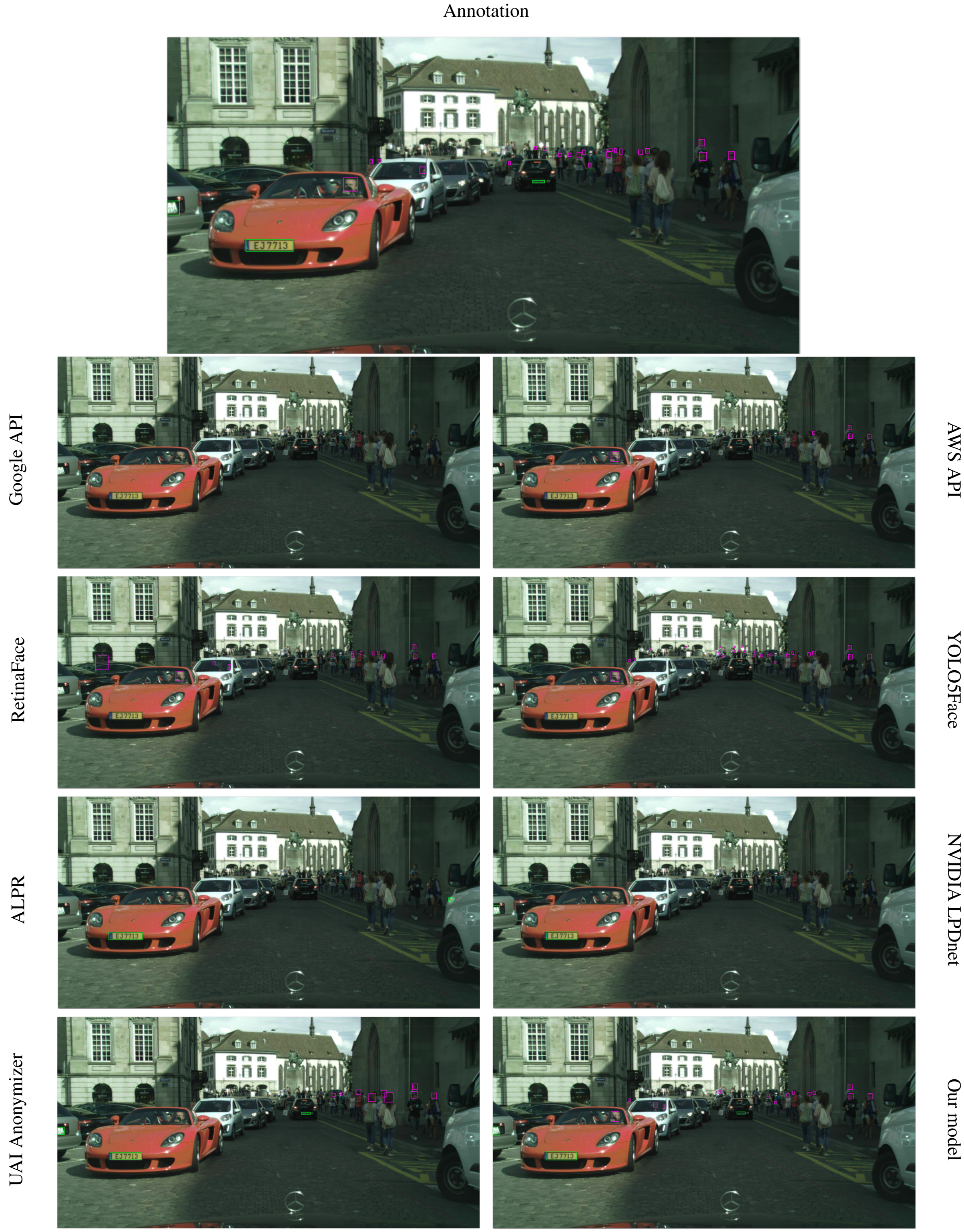


Figure 5: Example results for face and license plate detection on a normal camera image from the PP4AV dataset. The Google API did not detect any faces. AWS API detects some faces close to the dash camera. Retina face detection detects more distant faces than the AWS API, although some of the detected faces are inaccurate. YOLO5Face detects nearly every face in the area, both near and far, with the exception of one face behind the windshield. The UAI Anonymizer detects both the face and the back head, however the back head is not annotated. Our model detects all nearby and medium-distance faces, but misses the distant face. ALPR detects a single plate, while NVIDIA LPDnet detects two close plates. UAI Anonymizer and our model detect all three plates, whereas three plates are annotated.

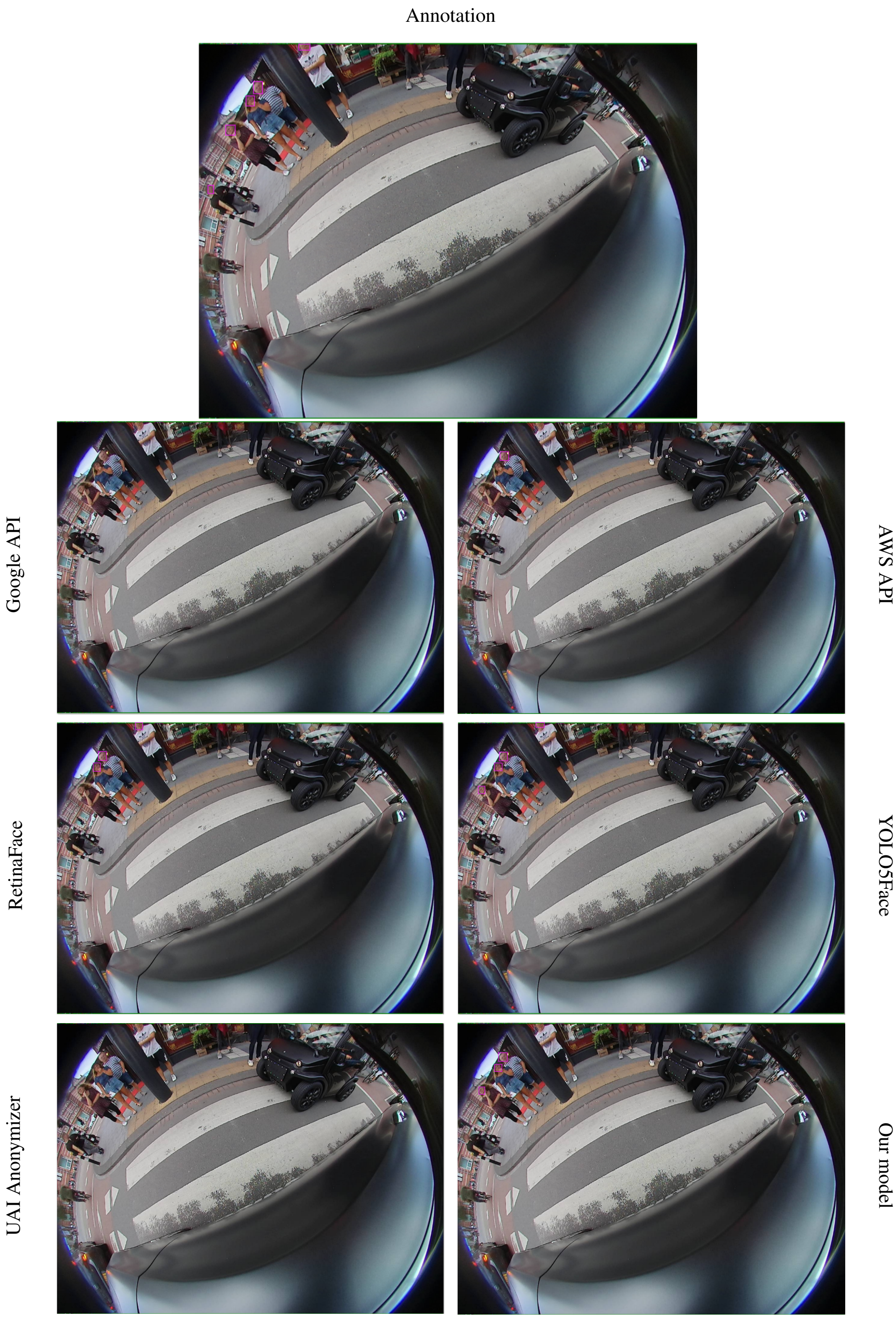


Figure 6: Example results for face detection on a fisheye image from the PP4AV dataset. UAI Anonymizer and Google API failed to detect any faces, however AWS detected one face more accurately. RetinaFace detected three faces, but missed 2 different faces and one obliged face. Our model detect three distinct faces, but misses two obligatory ones. Four faces are detected by YOLO5Face compared to five faces annotated.

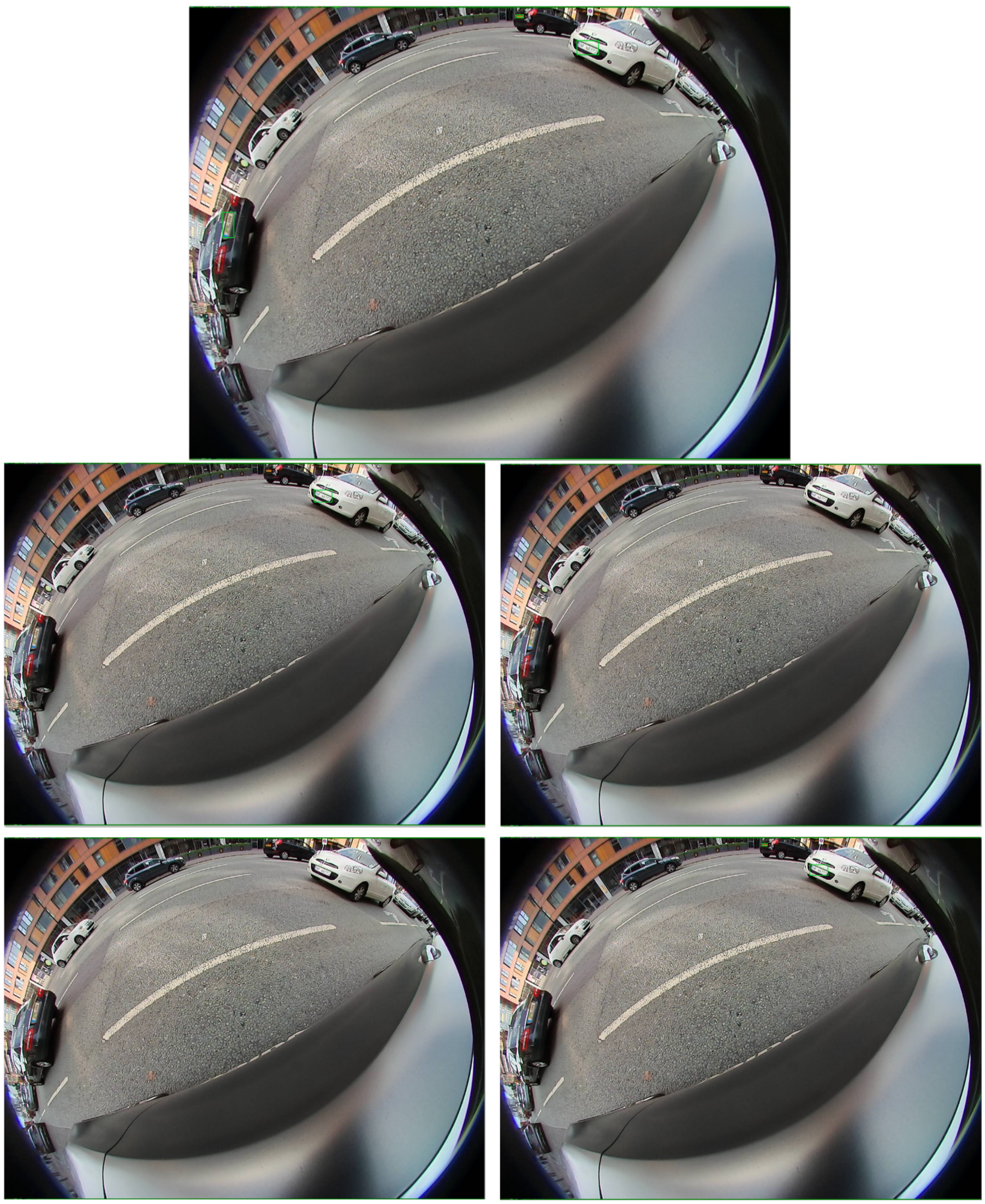


Figure 7: Example results for license plate detection on a fisheye image from the PP4AV dataset. UAI Anonymizer and NVIDA LPDnet had yet to detect a license plate, although ALPR did. We identify two plates across three plates of annotation as our model. Due to its obliqueness, none of the algorithms could detect the large plate on the left side of the image.